

I. REAL PARTY IN INTEREST	1
II. RELATED APPEALS AND INTERFERENCES.....	1
III. STATUS OF CLAIMS.....	2
IV. STATUS OF AMENDMENTS	2
V. SUMMARY OF CLAIMED SUBJECT MATTER.....	2
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	4
VII. THE ARGUMENT	4
VIII. CLAIMS APPENDIX	9
IX. EVIDENCE APPENDIX	13
X. RELATED PROCEEDINGS APPENDIX	14

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Application Number: 10/789,331
Filing Date: 2/27/2004
Applicant(s): Brian Levine
Entitled: APPLYING ORDERED MODIFICATIONS TO
RECURRING EVENT INSTANCES
Examiner: Tiphany B. Dickerson
Group Art Unit: 3623
Attorney Docket No.: LOT920040014US1 (7321-045U)

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is Appellant's Appeal Brief in support of the Notice of Appeal filed December 8, 2009. As this Appeal Brief has been timely filed within the shortened statutory period of two months from the date of the Notice of Appeal, no extension of time under 37 C.F.R. § 1.136 is required. Notwithstanding, please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, including extension of time fees, to Deposit Account 12-2158, and please credit any excess fees to such deposit account.

Date: February 8, 2010

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg, Registration No. 44,725
Customer Number 46321

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Application Number: 10/789,331
Filing Date: 2/27/2004
Applicant(s): Brian Levine
Entitled: APPLYING ORDERED MODIFICATIONS TO
RECURRING EVENT INSTANCES
Examiner: Tiphany B. Dickerson
Group Art Unit: 3623
Attorney Docket No.: LOT920040014US1 (7321-045U)

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed December 8, 2009, wherein Appellant appeals from the Examiner's rejection of claims 1 through 13.

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by assignment recorded on February 27, 2004, at Reel 015038, Frame 0478.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1 through 13 are pending in this Application. Claims 1 through 13 have been rejected twice. It is from the multiple rejections of claims 1 through 13 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Claims 1, 6, 7 and 9 were amended in the amendment dated May 4, 2009 (the "Amendment").

V. SUMMARY OF CLAIMED SUBJECT MATTER

As set forth in paragraph [0015] of the original specification (the "Specification"), Appellant has invented a system, method and apparatus for applying ordered modifications to recurring event instances. In accordance with an embodiment of Appellant's invention, an event storage model can include event data and a corresponding recurrence rule for each recurring event. Actual event instances can be calculated on demand based upon the recurrence rule. Notably, modifications to the event instances can be specified separately for specific event properties in the event instance. Only the modified properties can be stored as an "event exception". In this way, modifications to the event instances can be calculated concurrently with the dynamic calculation of the event instances based upon the recurrence rule. Finally, stale event exceptions which have become obviated by the application of subsequent event exceptions can be purged from the storage model periodically.

With specific respect to claim 1, a method for applying ordered modifications to recurring event instances is provided. (Par. [0019]) The method includes identifying an event

exception in a calendaring system executing in memory by a processor of a computer. (Par. [0019]) In this regard, the event exception corresponds to a separately defined and separately stored recurring event instance in the calendaring system. (Par. [0019]) Finally, the method includes modifying within the calendaring system at least one property of the recurring event instance based upon the separately stored event exception. (Par. [0020])

With specific respect to claim 6, a calendaring system is provided to include a computer with processor and memory and a recurrence event expander disposed within a calendaring system executing in the memory by the processor of the computer. (Par. [0016]) The expander is programmed to expand recurrence events into event instances based upon the properties specified within the recurrence events. (Par. [0016]) Also, a recurrence event modifier is disposed within the calendaring system and coupled to the recurrence event expander. (Par. [0017]) The modifier modifies properties within the event instances based upon event exceptions defined for respective ones of the event instances and separately stored from the event instances. (Par. [0017])

With specific respect to claim 9, a machine readable storage has stored thereon a computer program for applying ordered modifications to recurring event instances. (Par. [0024]) The computer program includes a routine set of instructions which when executed by a machine cause the machine to identify an event exception in a calendaring system executing in memory by a processor of a computer. (Pars. [0019], [0024]) In this regard, the event exception corresponds to a separately defined and separately stored recurring event instance in the

calendar system. (Par. [0019]) Finally, at least one property of the recurring event instance can be modified based upon the separately stored event exception. (Par. [0020])

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1 through 13 have been rejected under 35 U.S.C. § 112, first paragraph.

Claims 1 through 7 and 9 through 13 have been rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent Application Publication No. 2003/00614333 by Hall et al. (Hall).

Claim 8 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Hall in view of Examiner's "Official Notice".

VII. THE ARGUMENT

THE REJECTION OF CLAIMS 1 THROUGH 13 UNDER 35 U.S.C. § 112, FIRST PARAGRAPH

On page 3 of the final official action mailed August 5, 2009 (the "Final Office Action"), Examiner rejects claims 1 through 13 under 35 U.S.C. § 112, first paragraph. Specifically, Examiner states:

Claim 1-13 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. As pointed out in the Response to Arguments Section above, First, Examiner finds no support for the amended claim element, "separately stored recurring event instances," in Applicant's specification. Examiner suggests that all references to the unsupported matter be deleted or that specific evidence of support be provided.

Notwithstanding, Appellant refers the Honorable Board to paragraph [0015] of the Specification in which Appellant stated:

Notably, modifications to the event instances can specified separately for specific event properties in the event instance. Only the modified properties can be stored as an "event exception". In this way, modifications to the event instances can be calculated concurrently with the dynamic calculation of the event instances based upon the recurrence rule.

Thus, ample support has been provided in the Specification for the claimed teaching of "separately stored recurring event instances".

THE REJECTION OF CLAIMS 1 THROUGH 7 AND 9 THROUGH 13 UNDER 35 U.S.C. § 102

For the convenience of the Honorable Board, claims 2 through 5 stand or fall with claim 1, claims 7 and 9 stand or fall with claim 6 and claims 10 through 13 stand or fall with claim 9.

On page 4 of the Final Office Action, Examiner argues that Hall anticipates each of Appellant's claims 1 and 9. The factual determination of anticipation under 35 U.S.C. § 102 requires the identical disclosure, either explicitly or inherently, of each element of a claimed invention in a single reference.¹ Moreover, the anticipating prior art reference must describe the recited invention with sufficient clarity and detail to establish that the claimed limitations existed in the prior art and that such existence would be recognized by one having ordinary skill in the art.² Absence from an allegedly anticipating prior art reference of any claimed element negates anticipation.³

Claims 1, 6 and 9 each recite the important limitation of event exceptions defined for respective ones of the event instances and separately stored from the event instances. Exemplary claim 1 recites:

1. A method for applying ordered modifications to recurring event instances, the method comprising the steps of:

¹ In re Schreiber, 128 F.3d 1473, 1477 (Fed. Cir. 1997) ("To anticipate a claim, a prior art reference must disclose every limitation of the claimed invention, either explicitly or inherently"), In re Rijckaert, 9 F.3d 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989); Perkin-Elmer Corp. v. Computervision Corp., 732 F.2d 888, 894, 221 USPQ 669, 673 (Fed. Cir. 1984).

² See In re Spada, 911 F.2d 705, 708, 15 USPQ 1655, 1657 (Fed. Cir. 1990); Diversitech Corp. v. Century Steps Inc., 850 F.2d 675, 678, 7 USPQ2d 1315, 1317 (Fed. Cir. 1988).

³ Kloster Speedsteel AB v. Crucible, Inc., 793 F.2d 1565, 1571 (Fed. Cir. 1986)(emphasis added).

identifying an event exception in a calendaring system executing in memory by a processor of a computer, the event exception corresponding to a separately defined and separately stored recurring event instance in the calendaring system; and,
modifying within the calendaring system at least one property of said recurring event instance based upon said separately stored event exception.

Examiner on page 4 of the Final Office Action argues that Hall, paragraphs [0004], [0036] and [0037] provide just such a teaching of event exceptions that are separately stored from event instances. Specifically, Examiner argues:

Concerning claims 1 and 9, Hall discloses a method for applying ordered modifications to recurring event instances, the method comprising the steps of:

identifying an event exception in a calendaring system executing in memory by a processor of a computer, the event exception corresponding to a separately defined and separately stored recurring event instance in the calendaring system (Hall [4] and [37] as explained in the Response to Arguments Section; also, [36], wherein record tags are used to identify event exceptions); and,

modifying within the calendaring system at least one property of said recurring event instance based upon said separately stored event exception (Hall, [37], wherein the expander is used to modify records).

As previously noted by Appellant in the Amendment, paragraph [0004] of Hall provides a teaching opposite to the storage of event exceptions separately from event instances. Specifically, Appellant stated:

The teachings of paragraph [0004] of Hall provide an opposite teaching. For the convenience of Examiner, paragraph [0004] is reproduced as follows:

[0004] Where the shared data applications on each device use the same file formats, synchronization is a relatively simple process. However, modern computing environments tend to be more complex, often having devices from multiple manufacturers, each with differing file formats making it difficult, if not impossible, to accurately synchronize shared data. This is often the case with calendaring applications. Records representing the same event are often stored in different formats on different devices. More specifically, some calendaring applications have the capability of representing a recurring appointment with a single complex record. A complex record typically contains a general rule and one or more exceptions. A general rule identifies information such as a beginning date and an ending date for the recurring appointment, the days of the week, month, or year on which the appointment falls, and a time for each of those days. An exception is a variance from the general rule. Exceptions fall into two categories: deletions--when an appointment does not occur, for example, on a given day or for a given week; and modifications--when the time, location, or other data for an appointment on a given day is changed. For example, a child's soccer practice might occur every Tuesday and Thursday from 5:30 to 6:30 PM in the months of March and April. There are, however, exceptions--the starting time for the second Tuesday in April will be 4:30 PM instead of 5:30, and there is no practice on the first Thursday in May.

Thus, in Hall, both the exceptions to recurring appointments are stored in a single complex record and are not separately stored as required by each of Applicant's amended claims 1, 6 and 9.

In response, on page 3 of the Final Office Action, Examiner argued that paragraph [0004] stood only for the proposition that the event instances are defined separately from event exceptions, but not necessarily stored separately. However, Examiner asserted that paragraph [0037] addresses this deficiency. Specifically, Examiner argued:

6. Second, Hall [4] teaches that the general rule/event data and exceptions are separately defined by stating that "[a] complex record contains a general rule and one or more exceptions."
7. Finally, Hall further teaches that the event exception may be separately stored from corresponding event instances. Hall teaches that the record merely holds the general rule for the data and the event exceptions as evidenced in Hall Fig. 5. However, tags and a synch engine are used to expand the rules and exceptions into event instances which are synchronized across the shared data environment (Hall [37]).

For the convenience of the Honorable Board, paragraph [0037] of Hall provides:

[0037] FIG. 8 illustrates the logical components, expander 70 and retractor 72, of synch engine 20. When contained in a given data store 18, each record 28 is stored in a format native to a corresponding shared data application 16. Records 28 in different data stores 18, then, can be stored in any number of formats. As described above, a recurring event can be represented by a single complex record or by a series of discrete records. When changes are made to a complex record on one device 12, synch engine 20 must be capable of updating a series of discrete records representing the same recurring event on another device 12. Expander 70 represents generally any programming capable of expanding a single complex record into a series of discrete records according to the complex record's rule 34 and exceptions 36. For example, a complex record 28 may include a rule 34 indicating that a meeting is to be held every Monday and Wednesday for a two-month period at a particular time. The exceptions 36, may indicate that a time modification for the second Monday in the first month and a deletion or absence of an appointment on the first Wednesday in the second month. In this example, expander 70 would create a series of distinct records 28, one for each appointment indicated in the rule 34. Expander 70 would then delete the record created for the first Wednesday of the second month and modify the time for the record created for the second Monday of the first month.

Thus, the last three sentences of paragraph [0037] teach that an "expander" applies a rule and any exceptions to the rule to an existing event in order to generate resulting event instances as discrete records. In Appellant's claims, however, the event instances must be stored separately from any event exceptions for the event instances. The "event exceptions" of Hall, as evidenced by paragraph [0037] are not event exceptions for event instances, but exceptions to a recurrence rule for an event. In particular, paragraph [0015] of the Specification of Appellant makes clear, an event exception is an event instance amongst recurring events that has been modified whereas paragraph [0037] makes clear that that an "exception" is a deviation from a recurrence rule.

Therefore, to the extent that Examiner has not located in Hall this important teaching of an event exception that is separately stored from an event instance, Examiner has not established a prima facie case of anticipation under the law.

THE REJECTION OF CLAIM 8 UNDER 35 U.S.C. § 103

In as much as Examiner has not established a prima facie case of anticipation with respect to claim 6, Examiner cannot have established a prima facie case of obviousness and, at least for this reason, claim 8 is not unpatentable over Hall and Examiner's official notice.

Appellant, therefore, respectfully solicits the Honorable Board to reverse the Examiner's objections and the Examiner's rejections under 35 U.S.C. § 112, first paragraph and 102(b).

Date: February 9, 2010

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg
Registration No. 44,725
Customer Number 46321

VIII. CLAIMS APPENDIX

1. (Previously Amended) A method for applying ordered modifications to recurring event instances, the method comprising the steps of:

identifying an event exception in a calendaring system executing in memory by a processor of a computer, the event exception corresponding to a separately defined and separately stored recurring event instance in the calendaring system; and,

modifying within the calendaring system at least one property of said recurring event instance based upon said separately stored event exception.

2. (Original) The method of claim 1, further comprising the steps of:

further identifying additional event exceptions corresponding to said recurring event instance; and,

for each one of said further identified additional event exceptions, further modifying at least one property of said recurring event instance based upon said one of said further identified additional event exceptions.

3. (Original) The method of claim 1, further comprising the step of repeating said identifying and modifying steps for additional ones of the recurring event instances.

4. (Original) The method of claim 1, further comprising the steps of:

further identifying event exceptions relating to said recurring event instance which have become stale; and,

purging said further identified event exceptions.

5. (Original) The method of claim 4, wherein said further identifying step comprises the step of further identifying event exceptions whose specified modifications to event properties in said recurring event instance have been obviated by modifications specified in subsequently defined event exceptions.

6. (Previously Amended) A calendaring system comprising:
a computer with processor and memory;
a recurrence event expander disposed within a calendaring system executing in the memory by the processor of the computer, the expander being programmed to expand recurrence events into event instances based upon the properties specified within the recurrence events; and,
a recurrence event modifier also disposed within the calendaring system and coupled to said recurrence event expander, the modifier modifying properties within the event instances based upon event exceptions defined for respective ones of the event instances and separately stored from the event instances.

7. (Previously Amended) The system of claim 6, further comprising a modification rule reducer coupled to the recurrence event modifier, the modification rule reducer inspecting older ones of the event exceptions to detect stale event exceptions.

8. (Original) The system of claim 6, wherein the calendaring system implements an iCalendar.

9. (Previously Amended) A machine readable storage having stored thereon a computer program for applying ordered modifications to recurring event instances, the computer program comprising a routine set of instructions which when executed by a machine cause the machine to perform the steps of:

identifying an event exception in a calendaring system executing in memory by a processor of a computer, the event exception corresponding to a separately defined and separately stored recurring event instance in the calendaring system; and,

modifying within the calendaring system at least one property of said recurring event instance based upon said separately stored event exception.

10. (Original) The machine readable storage of claim 9, further comprising the steps of: further identifying additional event exceptions corresponding to said recurring event instance; and,

for each one of said further identified additional event exceptions, further modifying at least one property of said recurring event instance based upon said one of said further identified additional event exceptions.

11. (Original) The machine readable storage of claim 10, further comprising the step of repeating said identifying and modifying steps for additional ones of the recurring event instances.

12. (Original) The machine readable storage of claim 10, further comprising the steps of:
further identifying event exceptions relating to said recurring event instance which have become stale; and,
purging said further identified event exceptions.

13. (Original) The machine readable storage of claim 12, wherein said further identifying step comprises the step of further identifying event exceptions whose specified modifications to event properties in said recurring event instance have been obviated by modifications specified in subsequently defined event exceptions.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.